

REMARKS/ARGUMENTS

I. STATUS OF CLAIMS

Claims 1-26 remain in this application. Claims 1, 2, 6, 7, 8, 13, 14, 15, 20, 21, and 22 have been amended. Claims 27-31 have been added. No claims are canceled. No new matter has been added from the addition of Claims 27-31. Claims 27-31 are apparatus claims directed to similar subject matter as Claims 1-5.

II. CLAIM REJECTIONS – 35 U.S.C. § 103

The Office Action rejected Claims 1-26 under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 5,754,858 to Broman et al., in view of U.S. Patent Application Publication #2003/0217193 A1 to Thurston et al. The rejection is respectfully traversed.

Claims 1 provides a method of a development and build environment for packaged software delivery in a distributed network of nodes, and features **gathering application program interface (API) dependency information for the first module, wherein the first module can provide and use at least one API, by (a) receiving a list of dependent modules for a given process of DLL module; (b) storing, in the first module's metadata, dependency information for the dependent modules in the list wherein dependency information includes API names and version that the process of DLL depends on; (c) collecting additional dependency information documented in one or more additional modules specifications wherein, dependency information includes API and versions that the process or DLL depends on; and (d) storing the additional dependency information in the first module's metadata.**

Broman and Thurston taken alone or in combination does not teach, disclose, or suggest a system that includes gathering application program interface (API) dependency information for a first module, wherein the first module can provide and use at least one API, as recited in steps (a-

d). (The designations (a) through (d) are provided only for the purpose of efficient description in this paper, and not to indicate a particular order, limitation or restriction.)

Steps (a-d) generally how the build environment creates metadata and dependency data used by nodes for package and module dependency checks. Steps (a-d) recite how the API dependency data is derived and maintained. These steps are not found in the cited references. For example, **receiving a list of dependent modules for a given process of DLL module, and storing, in the first module's metadata, dependency information for the dependent modules in the list wherein dependency information includes API names and version that the process of DLL depends on**, are not disclosed, taught or suggested by Broman and Thurston taken alone or in combination.

The Office Action cites col. 8, lines 25-28 and 44-45 as suggestive of “dependency information.” However, Broman describes a user interface that offers options to a user. Col. 8, lines 21-36 states:

“Referring to FIG. 4, the generator user interface 78 presented by the custom application project generator 52 and the services module 76 also has the form of a dialog 150 with a title bar 152 at its top edge, a button bar 154 at its bottom edge, and an inner dialog 156 in between. The custom application project generator 52 and the services module 76 present a sequence of application project option selection steps in the inner dialog 156 which the user navigates by clicking on a back, next and finish buttons 160-162. The inner dialog 156 for each step contains a set of user interface controls for the user to choose from related application project options. The application project options and option selection steps presented in the dialog 150 are definable by the writer, and may include a mix of standard application project option steps and the writer's custom steps as described below.”

Applicants do not see how a user interface can suggest the claimed “dependency information.”

The user interface of Broman is irrelevant to gathering API dependency information by steps (a-d).

The Office Action additionally relies on Thurston’s disclosure of metadata [008-009] “to control installations and dependency information placed into ‘constraints’ that must be satisfied into a header,” citing FIG. 4. However, this portion of Thurston merely describes the contents of the metadata used by Thurston’s firmware update application. Specifically Thurston states,

“The metadata further comprises a header, where the header includes characteristics of a firmware update package that includes the firmware image. The metadata also comprises of dynamic constraints, wherein the firmware update application determines whether the dynamic constraints are satisfied before installing the firmware image on the hardware device.”

Nothing in Thurston discloses the claimed steps of **gathering application program interface (API) dependency information for the first module wherein the first module can provide and use at least one API, by receiving a list of dependent modules for a given process of DLL module; storing, in the first module’s metadata, dependency information for the dependent modules in the list wherein dependency information includes API names and version that the process of DLL depends on.**

Furthermore, nothing in either of the cited references discloses the claimed steps of **collecting additional dependency information documented in one or more additional modules specifications wherein, dependency information includes API and versions that the process or DLL depends on and storing the additional dependency information in the first module’s metadata.** Addressing the “collecting” step, the Office Action states:

“- collecting dependencies document in module specifications ...

Broman disclosed (col. 10, lines 30), application project options, suggestive of dependencies. Also, Broman disclosed (col. 11, lines 63-67) make files which list source does files that must be complied (considered dependencies) to build the application).”

However, the Office Action incorrectly correlates use of “application project options” and “make files” to **collecting dependencies document in the module specification**. In Claim 1 the dependency information includes APIs and versions that the process or DLL depends on. This is not a case of claim language so broad as to encompass the cited art—Broman has nothing equivalent.

The Office action further relies on Broman’s disclosure of template processing directives to inherently include API names and versions (col. 17 lines 1-24 and col. 8 lines 38-48). Broman states that “binary templates typically contain bitmap data for user interface components, such as toolbars and the like, which are copied verbatim by the services module 76 into the application project 54.” The Office Action’s characterization of bitmapped user interface components as APIs is completely contradictory to usage of those terms among skilled artisans. However, even assuming Broman’s “bitmap data” could be considered dependency information including APIs and versions that a process or DLL depends upon, nothing in Broman discloses **collecting the dependency information documented in one or more additional module specifications**.

Therefore, Broman in view of Thurston taken alone or in combination does not teach, disclose, or suggest the invention as claimed. Accordingly, Claim 1 is in allowable condition.

Independent Claims 6, 13, 20, and 27 are similarly allowable because they include the same features discussed above for Claim 1.

Claims 2-5, and 7-12, and 14-19, and 21-26, and 28-31 are dependent upon independent Claims 1, 6, 13, 20, and 27 respectively. By dependency, Claims 2-5, and 7-12, and 14-19, and 21-26, and 28-31 include the same features discussed above for Claim 1. Therefore, Claims 2-5,

and 7-12, and 14-19, and 21-26, and 28-31 are allowable for the same reasons discussed above for Claim 1.

For all the foregoing reasons, Applicants respectfully request reconsideration and withdrawal of the rejection under 35 U.S.C. §103(a).

III. CONCLUSION

The Applicants believe that all issues raised in the Office Action have been addressed and that allowance of the pending claims is appropriate. Entry of the amendments herein and further examination on the merits are respectfully requested.


The Examiner is invited to telephone the undersigned at (408) 414-1227 to discuss any issue that may advance prosecution.

No fee is believed to be due specifically in connection with this Reply. To the extent necessary, Applicants petition for an extension of time under 37 C.F.R. § 1.136. The Commissioner is authorized to charge any fee that may be due in connection with this Reply to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: June 7, 2006



Andrea L. Bauer
Reg. No. 58,438

2055 Gateway Place, Suite 550
San Jose, California 95110-1089
Telephone No.: (408) 414-1080 ext. 227
Facsimile No.: (408) 414-1076